Infrastructure sécurisée Linux

DATE
10/02/2025

Said HASHEMI
Intervenant : Mr Decker

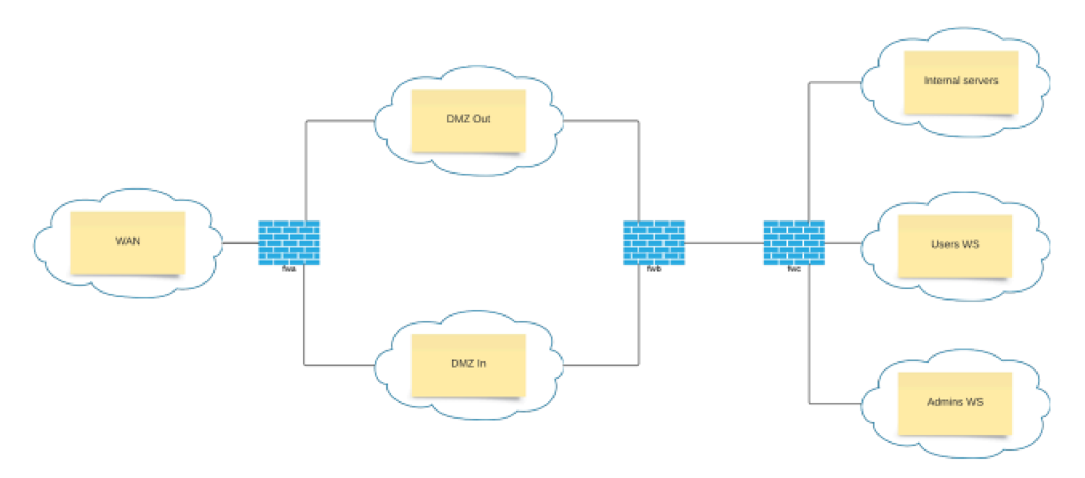
Objectif du projet

L'objectif de ce projet est de mettre en œuvre une infrastructure réseau sécurisée en suivant les spécifications détaillées dans le document fourni. L'architecture décrite inclut l'installation et la configuration de divers services sur une distribution Linux Debian 12, dans un environnement virtualisé utilisant VMWare Workstation ou VirtualBox. Les technologies clés à déployer comprennent le filtrage avec IPTables, la gestion DNS avec Bind9, le DHCP via Kea DHCP, le routage avec FRR Routing, un serveur web Apache2 ou NGINX, ainsi qu'un serveur de logs centralisé Syslog-NG ou RSyslog. De plus, un proxy Squid et un reverse proxy seront configurés pour assurer la sécurité et la gestion du trafic réseau. Chaque service devra être minutieusement configuré et sécurisé, avec une attention particulière à l'hardening des serveurs et des services implémentés, en suivant des guides de durcissement standards. Le livrable attendu comprend une preuve de concept détaillée, un plan d'action complet, des configurations de services, des justifications des choix techniques, des plans d'adressage, ainsi que des règles de filtrage et de validation des tests, dans un document de qualité professionnelle.

Table des matières

1. Schéma logique du réseau

Le schéma logique représente l'architecture globale du réseau avec ses différentes zones (WAN, DMZ In, DMZ Out, Internal Servers, Users WS, Admins WS) et les interconnexions entre les firewalls (**fwa**, **fwb**, **fwc**).



2. Choix de virtualisateur

J'ai choisi d'utiliser VMware comme virtualiseur pour ce projet, car il offre une bonne gestion des machines virtuelles et une stabilité de fonctionnement. J'ai ensuite téléchargé Debian 12 depuis le site officiel de Debian, en sélectionnant la version sans environnement de bureau et XFCE. Cette option permet de réduire la consommation des ressources, de garder un environnement minimal et de mieux gérer les services réseau, ce qui est adapté pour un serveur ou une infrastructure de réseau sécurisée. Cela permet également d'éviter la surcharge liée à l'interface graphique, qui n'est pas nécessaire pour ce type de configuration.

3. Adressage des équipements intermédiaires

Composant	**Interface**	**Adresse IP**	**Description**	**Passerelle**
fwa	Ens33	DHCP	WAN	Auto
	Ens34	10.0.0.1/11	DMZ In	
	Ens35	10.32.0.1/11	DMZ Out	
fwb	Ens33	10.128.0.1/30	Vers fwc	
	Ens34	10.0.0.2/11	DMZ In	
	Ens35	10.32.0.2/11	DMZ Out	
fwc	Ens37	10.128.0.2/30	Vers fwb	
	Ens34	10.64.0.1/12	Internal Servers	
	Ens35	10.80.0.1/12	Users WS	
	Ens36	10.96.0.1/12	Admins WS	

3.1 Configurations des cartes réseaux

J'ai configuré les cartes réseau sur trois pare-feu pour segmenter le réseau de manière sécurisée. Sur **FWA**, l'interface **Ens33** est en DHCP pour se connecter au WAN, **Ens34** est configurée pour la DMZ In (10.0.0.1/11), et **Ens35** pour la DMZ Out (10.32.0.1/11). Sur **FWB**, **Ens33** est reliée à **FWC** (10.128.0.1/30), **Ens34** et **Ens35**

gèrent la DMZ In et Out respectivement (10.0.0.2/11 et 10.32.0.2/11). Sur **FWC**, **Ens37** est connectée à **FWB** (10.128.0.2/30), **Ens34** gère les serveurs internes (10.64.0.1/12), **Ens35** les postes utilisateurs (10.80.0.1/12), et **Ens36** les postes administrateurs (10.96.0.1/12).

Sur FWC :

```
# Configuration pour enp0s3 - Lien vers fwb
iface ens37 inet static
    address 10.128.0.2
    netmask 255.255.255.252
    gateway 10.128.0.1
# Configuration pour enp0s8 - DMZ interne (serveurs internes)
iface ens34 inet static
    address 10.64.0.1
    netmask 255.240.0.0

# Configuration pour enp0s9 - DMZ pour les utilisateurs (workstations)
iface ens35 inet static
    address 10.80.0.1
    netmask 255.240.0.0

# Configuration pour enp0s10 - Réseau Administrateurs
iface ens36 inet static
    address 10.96.0.1
    netmask 255.240.0.0
```

Sur FWB :

```
auto ens33
iface ens33 inet static
  address 10.128.0.1/30

auto ens34
iface ens34 inet static
  address 10.0.0.2/11

auto ens35
iface ens35 inet static
  address 10.32.0.2/11
```

Sur FWA :

```
root@fwa:/home/admin-infra# cat /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback
auto ens33
iface ens33 inet dhcp
auto ens34
iface ens34 inet static
address 10.0.0.1/11
auto ens35
iface ens35 inet static
address 10.32.0.1/11
```

4. Routes

J'ai configuré des routes pour chaque pare-feu afin de gérer le trafic entre les différentes zones du réseau :

- **FWA** : Connecte le WAN en DHCP, la DMZ d'entrée (10.0.0.1) et la DMZ de sortie (10.32.0.1).
- **FWB** : Relie FWA et FWC via une connexion point-à-point (10.128.0.1), et gère la DMZ d'entrée (10.0.0.2) et de sortie (10.32.0.2).
- **FWC** : Relie FWB avec l'adresse 10.128.0.2, et gère les serveurs internes (10.64.0.1), les postes utilisateurs (10.80.0.1) et les postes administrateurs (10.96.0.1).

Sur FWC :

```
# routes
up ip route add 10.0.0.0/11 via 10.128.0.1 dev ens37
up ip route add 10.32.0.0/11 via 10.128.0.1 dev ens37
```

Sur FWB :

```
# Ajout des routes statiques
post-up ip route add 10.64.0.0/12 via 10.128.0.2 dev ens33
post-up ip route add 10.80.0.0/12 via 10.128.0.2 dev ens33
post-up ip route add 10.96.0.0/12 via 10.128.0.2 dev ens33

source /etc/network/interfaces.d/*
```

Sur FWA :

```
up ip route add 10.64.0.0/12 via 10.32.0.2 dev ens35
up ip route add 10.80.0.0/12 via 10.32.0.2 dev ens35
up ip route add 10.96.0.0/12 via 10.32.0.2 dev ens35
up ip route add 10.128.0.0/30 via 10.32.0.2 dev ens35
```

3.1 Routes avec FRRouting

J'ai installé **FRRouting**. Ensuite, j'ai activé **ospfd** (OSPF Daemon) pour activer le routage OSPF.

Sur FWC :

```
fwc# show ip ospf route
============ OSPF network routing table ============
N    10.64.0.0/12          [100] area: 0.0.0.0
                           directly attached to ens34
N    10.80.0.0/12          [100] area: 0.0.0.0
                           directly attached to ens35
N    10.96.0.0/12          [100] area: 0.0.0.0
                           directly attached to ens36
N    10.128.0.0/30         [100] area: 0.0.0.0
                           directly attached to ens37

============ OSPF router routing table =============

============ OSPF external routing table ===========
```

Sur FWB :

```
fwb# sh ip ospf route
============ OSPF network routing table ============
N    10.0.0.0/11           [100] area: 0.0.0.0
                           directly attached to ens34
N    10.32.0.0/11          [100] area: 0.0.0.0
                           directly attached to ens35
N    10.128.0.0/30         [100] area: 0.0.0.0
                           directly attached to ens33

============ OSPF router routing table =============

============ OSPF external routing table ===========
```

Sur FWA :

```
fwa# sh ip ospf route
============ OSPF network routing table ============
N    10.0.0.0/11           [100] area: 0.0.0.0
                           directly attached to ens34
N    10.32.0.0/11          [100] area: 0.0.0.0
                           directly attached to ens35

============ OSPF router routing table =============

============ OSPF external routing table ===========
```

5. Filtrage IP Tables

Après l'installation d'iptables, j'ai entré ces commandes sur les serveurs **fwc**, **fwb**, et **fwa** pour configurer les règles de filtrage :

1. **Sur FWC** :
   - J'ai configuré les règles permettant le passage du trafic entre les différentes interfaces réseau, en autorisant le trafic entre la DMZ et les réseaux internes (serveurs, utilisateurs et administrateurs).
   - J'ai spécifié des règles pour autoriser l'accès à Internet pour les utilisateurs tout en restreignant l'accès des utilisateurs aux réseaux d'administrateurs.
   - J'ai autorisé l'accès aux services nécessaires comme le reverse proxy Nginx, Bind9, Squid, et les services DHCP et Syslog.
2. **Sur FWB** :
   - J'ai configuré les règles pour permettre le passage du trafic entre les interfaces réseau, autorisant ainsi les connexions entre les réseaux DMZ-In et DMZ-Out.
   - J'ai appliqué des règles spécifiques pour permettre la communication entre FWB et FWC tout en restreignant l'accès direct aux autres services non autorisés.
3. **Sur FWA** :
   - J'ai mis en place des règles pour filtrer le trafic entrant en fonction des besoins du WAN, tout en autorisant les connexions nécessaires aux autres interfaces.
   - J'ai autorisé le trafic entre le réseau externe (WAN) et le DMZ-Out, puis vers FWB, conformément à l'architecture réseau définie.

Les règles identiques sur tous les FW :

```
iptables -F
iptables -X
iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -P OUTPUT DROP
iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT
```

Sur FWA :

```
iptables -A FORWARD -i ens33 -o ens35 -d 10.32.0.0/11 -j ACCEPT
iptables -A FORWARD -i ens34 -o ens33 -d 10.0.0.2/11 -j ACCEPT
```

Ensuite sauvegardé.

```
iptables-save > /etc/iptables/rules.v4
```

Sur FWB :

```
iptables -A FORWARD -i ens33 -o ens33 -d 10.128.0.2/30 -j ACCEPT
iptables -A FORWARD -i ens34 -o ens35 -d 10.32.0.0/11 -j ACCEPT
```

Sur FWA :

```
iptables -A FORWARD -i ens37 -o ens33 -d 10.128.0.1/30 -j ACCEPT
iptables -A FORWARD -i ens37 -o ens34 -d 10.64.0.0/12 -j ACCEPT
iptables -A FORWARD -i ens37 -o ens35 -d 10.80.0.0/12 -j ACCEPT
iptables -A FORWARD -i ens37 -o ens36 -d 10.96.0.0/12 -j ACCEPT
iptables -A FORWARD -i ens36 -o ens33 -j ACCEPT
iptables -A FORWARD -i ens36 -o ens34 -d 10.64.0.0/12 -j ACCEPT
iptables -A FORWARD -i ens35 -o ens36 -d 10.96.0.0/12 -j DROP
iptables -A INPUT -p tcp --dport 3128 -j ACCEPT
iptables -A INPUT -p tcp --dport 80 -j ACCEPT
iptables -A INPUT -p tcp --dport 443 -j ACCEPT
iptables -A INPUT -p udp --dport 53 -j ACCEPT
iptables -A INPUT -p tcp --dport 53 -j ACCEPT
iptables -A INPUT -p udp --dport 67 -j ACCEPT
iptables -A INPUT -p udp --dport 68 -j ACCEPT
iptables -A INPUT -p udp --dport 514 -j ACCEPT
iptables -A INPUT -j DROP
iptables -A FORWARD -j DROP
iptables -A OUTPUT -j ACCEPT
```

Les règles sont sauvgardé sur chaque FW avec la commande iptables-save > /etc/iptables/rules.v4 et aussi en utilisant iptables-persistent pour restaurer les règles au démarrage.

6. DNS et DHCP

J'ai installé **Bind9** (le serveur DNS) et **DHCP** sur une machine avec l'adresse IP 10.64.0.1. J'ai choisi de les installer sur cette machine pour optimiser l'utilisation de la mémoire de mon hôte, car ma machine physique dispose de 8 Go de RAM, ce qui limite le nombre de machines virtuelles que je peux déployer. En centralisant ces services sur une seule machine, je peux réduire la charge et améliorer la gestion des ressources de l'hôte.

```
root@fwc:~# systemctl status bind9
● named.service - BIND Domain Name Server
     Loaded: loaded (/lib/systemd/system/named.service; enabled; preset: enabled)
     Active: active (running) since Mon 2025-02-10 10:28:55 CET; 3h 19min ago
       Docs: man:named(8)
   Main PID: 2227 (named)
     Status: "running"
      Tasks: 8 (limit: 1035)
     Memory: 11.5M
        CPU: 978ms
     CGroup: /system.slice/named.service
             └─2227 /usr/sbin/named -f -u bind

févr. 10 12:51:31 fwc named[2227]: automatic empty zone: 8.B.D.0.1.0.0.2.IP6.ARPA
févr. 10 12:51:31 fwc named[2227]: automatic empty zone: EMPTY.AS112.ARPA
févr. 10 12:51:31 fwc named[2227]: automatic empty zone: HOME.ARPA
févr. 10 12:51:31 fwc named[2227]: automatic empty zone: RESOLVER.ARPA
févr. 10 12:51:31 fwc named[2227]: configuring command channel from '/etc/bind/rndc.key'
févr. 10 12:51:31 fwc named[2227]: configuring command channel from '/etc/bind/rndc.key'
févr. 10 12:51:31 fwc named[2227]: reloading configuration succeeded
févr. 10 12:51:31 fwc named[2227]: scheduled loading new zones
févr. 10 12:51:31 fwc named[2227]: any newly configured zones are now loaded
févr. 10 12:51:31 fwc named[2227]: running
```

```
root@fwc:~# systemctl status isc-dhcp-server
● isc-dhcp-server.service - LSB: DHCP server
     Loaded: loaded (/etc/init.d/isc-dhcp-server; generated)
     Active: active (running) since Mon 2025-02-10 10:33:46 CET; 3h 16min ago
       Docs: man:systemd-sysv-generator(8)
      Tasks: 1 (limit: 1035)
     Memory: 3.2M
        CPU: 89ms
     CGroup: /system.slice/isc-dhcp-server.service
             └─2832 /usr/sbin/dhcpd -4 -q -cf /etc/dhcp/dhcpd.conf ens34

févr. 10 10:33:44 fwc systemd[1]: Starting isc-dhcp-server.service - LSB: DHCP server...
févr. 10 10:33:44 fwc isc-dhcp-server[2819]: Launching IPv4 server only.
févr. 10 10:33:44 fwc dhcpd[2832]: Wrote 0 leases to leases file.
févr. 10 10:33:44 fwc dhcpd[2832]: Server starting service.
févr. 10 10:33:46 fwc isc-dhcp-server[2819]: Starting ISC DHCPv4 server: dhcpd.
févr. 10 10:33:46 fwc systemd[1]: Started isc-dhcp-server.service - LSB: DHCP server.
```

6.1 Installation de Bind9 (DNS) :

Bind9 est un serveur DNS qui permet de résoudre les noms de domaine en adresses IP. Il a été installé pour gérer la résolution des noms de domaine au sein du réseau interne.

6.2 Installation de DHCP :

Le serveur DHCP a été installé pour attribuer automatiquement les adresses IP aux clients du réseau interne. Cela simplifie la gestion des adresses IP pour les différents équipements sans nécessiter de configuration manuelle.

6.7 Test de DNS et DHCP

Sur la VM « interne » la carte ens34 a bien pris son IP du serveur DHCP installé sur la machine ayant l'adresse IP 10.64.0.1.

```
3: ens34: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state
up default qlen 1000
    link/ether 00:0c:29:6c:e9:44 brd ff:ff:ff:ff:ff:ff
    altname enp2s2
    inet 10.64.10.100/12 brd 10.64.255.255 scope global dynamic ens34
       valid_lft 594sec preferred_lft 594sec
```

Ci-dessous, en utilisant la commande « nslookup » j'ai interrogé le DNS du domaine « Said-Ynov » a chercher la machine « client1.said-ynov), voici le retour :

```
root@interne:~# nslookup client1.said-ynov
Server:         10.64.0.1
Address:        10.64.0.1#53

Name:   client1.said-ynov
Address: 10.80.0.10
```

7. Proxy Squid

J'ai installé **Proxy Squid** sur **FWC**, car cette machine contient le réseau interne, et l'utilisation d'un proxy permet de gérer et de filtrer le trafic web sortant des utilisateurs internes vers Internet. Le proxy **Squid** est particulièrement utile pour :

- **Améliorer la sécurité** en filtrant les connexions sortantes et en enregistrant les logs des requêtes HTTP.
- **Optimiser la bande passante** en mettant en cache les pages web fréquemment consultées, réduisant ainsi le besoin de récupérer ces données à chaque fois qu'elles sont demandées.
- **Appliquer des politiques de filtrage** pour autoriser ou bloquer l'accès à certains sites en fonction des règles de sécurité.

L'installation de Squid sur **FWC** permet de centraliser la gestion du trafic web des utilisateurs internes tout en sécurisant l'accès à Internet.

```
root@fwc:~# dpkg -l | grep squid
ii  squid                          5.7-2+deb12u2
ii  squid-common                   5.7-2+deb12u2
ii  squid-langpack                 20220130-1
```

7.1 Test de proxy Squid

J’ai utilisé Curl pour tester Squid, voici le retour de la commande :
Curl -x http://10.128.0.2:3128 -L https://www.google.com

```
root@interne:~# curl -x http://10.128.0.2:3128 -L https://www.google.com
<!doctype html><html itemscope="" itemtype="http://schema.org/WebPage" lang="fr"><head><meta content="text/html; charset=UTF-8" http-equiv="Content-Type"><meta content="/images/branding/goo
leg/1x/googleg_standard_color_128dp.png" itemprop="image"><title>Google</title><script nonce="Rc3E7x2cMKiFev3t53wU4Q">(function(){var _g={kEI:'hBKqZ6W5BdGC7M8P1P776AU',kEXPI:'0,3700260,689,
35,538661,2872,2891,43028,30022,16105,344796,94243,117808,47082,340,30571,5240757,766,8835014,84,7437948,20539755,25228681,78017,34189,10898,15164,8181,5937,4556,3894,56130,585,6751,23879,9
40,4598,328,6226,7974,1988,24348,3511,26344,1341,13708,8213,7421,3750,10771,3048,239,3494,33,26708,10671,11736,6764,3,2,5823,1498,3854,41,13639,1,1443,4095,1203,2895,1210,302,4156,3887,1583
,2,919,951,2146,4617,5773,4310,2327,1308,2,738,7967,9171,1769,20,3661,5962,4863,2038,3261,41,2,415,4024,637,172,6,359,789,714,639,2,209,918,1275,55,525,281,218,1,319,328,183,1,218,1,703,193
,222,919,532,5,74,3207,570,2367,423,1,370,1059,1123,143,452,9,382,1152,35,1273,217,1223,955,354,1029,15,139,3,2810,4,5,404,810,541,14,8,826,6,1553,900,1009,342,12,362,351,687,3,2,1,2,2,2,3,
88,189,266,214,16,175,121,141,323,286,291,311,57,158,3,818,394,497,11,1,346,21,169,361,1112,273,2,735,581,1046,124,973,476,123,154,437,868,413,74,252,354,745,426,1075,311,101,3,463,119,16,2
0,24,260,46,105,16,205,2,37,129,307,22,6,60,440,164,293,8,1,1,3,569,44,1003,39,1011,151,153,92,141,388,173,42,461,96,8,862,28,32,55,342,10,211,5,21332160,4,28852,18,2013,40,2487,8,3873,12,2
3,4364,550,8018905',kBL:'9Rrr',kOPI:89978449};(function(){var a;((a=window.google)==null?0:a.stvsc)?google.kEI=_g.kEI:window.google=_g;}).call(this);})();(function(){google.sn='webhp';googl
.kHL='fr';})();(function(){
var g=this||self;function k(){return window.google&&window.google.kOPI||null};var l,m=[];function n(a){for(var b;a&&(!a.getAttribute||!(b=a.getAttribute("eid")));)a=a.parentNode;return b||l
function p(a){for(var b=null;a&&(!a.getAttribute||!(b=a.getAttribute("leid")));)a=a.parentNode;return b}function q(a){/^http:/i.test(a)&&window.location.protocol==="https:"&&(google.ml&&goo
le.ml(Error("a"),!1,{src:a,glmm:1}),a="");return a}
```

8. Reverse Proxy Nginx

J'ai installé **Nginx** en tant que **Reverse Proxy** sur **FWC**, car cela permet de gérer et de sécuriser les accès entrants vers les serveurs internes, tout en répartissant le trafic de manière optimale. Le **Reverse Proxy** fonctionne en tant qu'intermédiaire entre les utilisateurs externes et les services internes, permettant ainsi de :

- **Masquer les serveurs internes** pour les rendre invisibles aux utilisateurs externes.
- **Répartir la charge** en dirigeant les requêtes vers plusieurs serveurs internes selon des règles de distribution de trafic.
- **Améliorer la sécurité** en filtrant et en limitant les accès aux serveurs internes, réduisant ainsi les risques d'attaques directes.
- **Optimiser les performances** en gérant la mise en cache des ressources demandées fréquemment et en réduisant la charge sur les serveurs internes.

L'installation de **Nginx** en tant que **Reverse Proxy** sur **FWC** permet de renforcer la sécurité tout en assurant une gestion fluide du trafic web entrant vers les applications et services hébergés en interne.

```
root@fwc:~# dpkg -l | grep nginx
ii  nginx                          1.22.1-9
ii  nginx-common                   1.22.1-9
```

8.1 Test de Reverse proxy

```
root@fwc:~# curl -I http://localhost
HTTP/1.1 200 OK
Server: nginx/1.22.1
Date: Mon, 10 Feb 2025 15:19:34 GMT
Content-Type: text/html
Content-Length: 615
Last-Modified: Mon, 10 Feb 2025 10:21:37 GMT
Connection: keep-alive
ETag: "67a9d331-267"
Accept-Ranges: bytes
```

9. Syslog

J'ai mis en place le **serveur de logs centralisé** (Syslog-NG ou RSyslog) sur **FWA**, qui est exposé à l'internet. Cependant, pour des raisons de sécurité et de gestion des ressources, j'ai choisi de sauvegarder les fichiers de logs dans le réseau **DMZ-Out** afin d'éviter toute compromission des logs par des attaques venant de l'extérieur.
Étant donné que la RAM de mon ordinateur hôte ne permet pas de supporter un grand nombre de machines virtuelles (j'ai déjà 5 VMs en fonctionnement), j'ai décidé d'installer ce service sur une machine déjà existante plutôt que de déployer une nouvelle machine virtuelle. Cette approche me permet d'optimiser l'utilisation des ressources de mon hôte sans surcharger sa capacité de mémoire.

9.1 Installation et configuration de Syslog :

1. **Installation de Syslog-NG ou RSyslog** : J'ai installé le service **Syslog-NG** (ou **RSyslog**) pour centraliser les logs de tous les serveurs de l'infrastructure. Ce service permet de collecter et d'enregistrer les logs provenant de différentes sources pour les analyser et les stocker de manière centralisée.
2. **Configuration du fichier Syslog** : Une fois Syslog-NG ou RSyslog installé, j'ai configuré le fichier de configuration de Syslog pour qu'il enregistre les logs des services et événements importants. J'ai spécifié le chemin des fichiers de log dans la **DMZ-Out**, en veillant à bien configurer les règles pour rediriger et archiver les logs provenant des différentes machines.
3. **Sécurisation des fichiers de log** : Pour garantir l'intégrité des logs, j'ai sécurisé les fichiers de configuration et les fichiers de log eux-mêmes. En utilisant des permissions strictes et des outils comme **chmod**, j'ai empêché toute suppression ou modification non autorisée de ces fichiers. Cela permet de garantir la traçabilité et de prévenir toute tentative de manipulation des logs en cas d'incident de sécurité.

Voici les commandes pour sécuriser les fichiers de log :
chmod 600 /var/log/syslog
chown syslog:adm /var/log/syslog
Ces commandes garantissent que seuls les utilisateurs autorisés (comme l'utilisateur **syslog**) peuvent accéder et modifier les fichiers de log, tandis que les autres utilisateurs ne peuvent pas les supprimer ou les modifier.
En résumé, j'ai centralisé les logs sur **FWA** tout en veillant à leur sécurité et à leur gestion efficace, et ce, malgré les contraintes de mémoire de mon hôte physique.

```
root@fwa:/home/admin-infra# dpkg -l | grep syslog-ng
ii  syslog-ng                          3.38.1-5                            all
ii  syslog-ng-core                     3.38.1-5                            amd64
ii  syslog-ng-mod-add-contextual-data  3.38.1-5                            amd64
ii  syslog-ng-mod-amqp                 3.38.1-5                            amd64
ii  syslog-ng-mod-examples             3.38.1-5                            amd64
ii  syslog-ng-mod-geoip2               3.38.1-5                            amd64
ii  syslog-ng-mod-graphite             3.38.1-5                            amd64
ii  syslog-ng-mod-http                 3.38.1-5                            amd64
ii  syslog-ng-mod-mongodb              3.38.1-5                            amd64
ii  syslog-ng-mod-python               3.38.1-5                            amd64
ii  syslog-ng-mod-rdkafka              3.38.1-5                            amd64
ii  syslog-ng-mod-redis                3.38.1-5                            amd64
ii  syslog-ng-mod-riemann              3.38.1-5                            amd64
ii  syslog-ng-mod-slog                 3.38.1-5                            amd64
ii  syslog-ng-mod-smtp                 3.38.1-5                            amd64
ii  syslog-ng-mod-snmp                 3.38.1-5                            amd64
ii  syslog-ng-mod-sql                  3.38.1-5                            amd64
ii  syslog-ng-mod-stardate             3.38.1-5                            amd64
ii  syslog-ng-mod-stomp                3.38.1-5                            amd64
ii  syslog-ng-mod-xml-parser           3.38.1-5                            amd64
ii  syslog-ng-scl                      3.38.1-5                            all
root@fwa:/home/admin-infra#
```

9.2 Test de Syslog

Au titre d'un exemple, logs des services de système :

```
Oct  5 10:55:01 debian dbus-daemon[1234]: [system] Successfully activated service 'or
g.freedesktop.hostname1'
Oct  5 10:56:22 debian NetworkManager[1235]: <info>  [1696503382.1234] device (eth0):
state change: ip-config -> ip-check (reason 'none')
```

Logs du noyau :

```
Oct  5 10:22:33 debian kernel: [ 2345.678901] ata1.00: exception Emask 0x0 SAct 0x0 S
Err 0x0 action 0x0
Oct  5 10:22:33 debian kernel: [ 2345.678902] ata1.00: irq_stat 0x40000001
Oct  5 10:22:33 debian kernel: [ 2345.678903] ata1.00: failed command: READ DMA
```

10. Serveur Web APACHE 2

J'ai installé **Apache2** sur une machine virtuelle avec l'adresse IP **10.64.0.100/12**. Cette machine a été placée dans la zone **Internal Servers**, car elle héberge un serveur web qui doit être accessible uniquement depuis le réseau interne pour des raisons de sécurité. Cela permet de limiter l'exposition des services aux seules machines autorisées. Cependant, si ce serveur Apache doit être accessible depuis l'extérieur (par exemple, pour des besoins d'accès public ou via un reverse proxy), il serait préférable de le déplacer dans la zone **DMZ In** sur **FWB**. La DMZ (zone

démilitarisée) est une zone intermédiaire entre le réseau interne et l'extérieur, permettant d'exposer certains services tout en les isolant du réseau interne pour réduire les risques d'attaque.

```
root@interne:~# sudo systemctl status apache2
• apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; enabled; preset: enabled)
     Active: active (running) since Mon 2025-02-10 16:00:34 CET; 12min ago
       Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 3986 (apache2)
      Tasks: 55 (limit: 1035)
     Memory: 8.9M
        CPU: 107ms
     CGroup: /system.slice/apache2.service
             ├─3986 /usr/sbin/apache2 -k start
             ├─3987 /usr/sbin/apache2 -k start
             └─3988 /usr/sbin/apache2 -k start

févr. 10 16:00:34 interne systemd[1]: Starting apache2.service - The Apache HTTP Server...
févr. 10 16:00:34 interne systemd[1]: Started apache2.service - The Apache HTTP Server.
```

10.1 Test Serveur Web

